

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

PAT-NO: JP408179937A
DOCUMENT-IDENTIFIER: JP 08179937 A
TITLE: DEVICE AND METHOD FOR SWITCHING BOOT PROGRAM
PUBN-DATE: July 12, 1996

INVENTOR-INFORMATION:
NAME
HONMA, HIDEO

ASSIGNEE-INFORMATION:
NAME COUNTRY
CANON INC N/A

APPL-NO: JP06324423
APPL-DATE: December 27, 1994

INT-CL (IPC): G06F009/06, G06F011/14

ABSTRACT:

PURPOSE: To provide a boot program switching device/method which can automatically switch plural boot programs without bothering an operator.

CONSTITUTION: Plural boot programs which boot a CPU 101 are stored in the ROM 104a to 104d respectively. The program stored in the ROM 104a is used for the ordinary booting operations. When a reset switch 106 is pushed, a reset part 102 resets the CPU 101 and also changes the address of the ROM to be asserted by an address decoder. Therefore, the internal address of the boot program contained in the CPU 101 and the external address of the actually corresponding ROM are converted. As a result, the boot program to be started is switched.

COPYRIGHT: (C)1996,JPO

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-179937

(43) 公開日 平成8年(1996)7月12日

(51) Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/06	4 1 0 V			
11/14	3 1 0 L			

審査請求 未請求 請求項の数17 O L (全 8 頁)

(21) 出願番号 特願平6-324423

(22) 出願日 平成6年(1994)12月27日

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 本間 英雄

東京都大田区下丸子3丁目30番2号 キヤ
ノン株式会社内

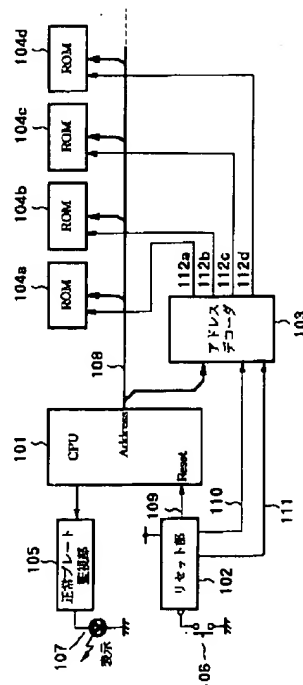
(74) 代理人 弁理士 大塚 康德 (外1名)

(54) 【発明の名称】 ブートプログラム切替え装置及びその方法

(57) 【要約】

【目的】 複数のブートプログラムを操作者の手を煩わせることなく、自動的に切替可能なブートプログラム切替え装置及びその方法を提供することを目的とする。

【構成】 CPU 101 をブートする複数のブートプログラムが、ROM 104 a ~ 104 d にそれぞれ格納されており、ROM 104 a に格納されているものが通常ブート用である。リセットスイッチ 106 が押下されると、リセット部 102 は CPU 101 をリセットするとともに、アドレスデコーダがアサートする ROM のアドレスを変更する。従って、CPU 101 の有するブートプログラムの内部アドレスと、実際に対応する ROM 上の外部アドレスとが変換され、結果として起動されるブートプログラムを切り替えることができる。



【特許請求の範囲】

【請求項1】 CPUをブートするためのブートプログラムを複数保持する保持手段と、

前記CPUの強制リセットを行うリセット手段と、
前記リセット手段によるリセットのタイミングで、前記保持手段に保持された複数のブートプログラムを切り替える切替え手段とを有することを特徴とするブートプログラム切替え装置。

【請求項2】 前記保持手段に保持された複数のブートプログラムのうち、1つは通常使用される既定ブートプログラムであることを特徴とする請求項1記載のブートプログラム切替え装置。

【請求項3】 前記切替え手段は、前記CPUにおけるブートプログラムの論理アドレスに前記保持手段における物理アドレスを割り当てることを特徴とする請求項1記載のブートプログラム切替え装置。

【請求項4】 前記保持手段はROMであることを特徴とする請求項1記載のブートプログラム切替え装置。

【請求項5】 前記保持手段は複数のROMであり、各ROM毎に1つのブートプログラムを保持することを特徴とする請求項4記載のブートプログラム切替え装置。

【請求項6】 前記切替え手段は、前記複数のROMのいずれかを特定する信号を出力し、該信号を切り替えることにより複数のブートプログラムを切り替えることを特徴とする請求項5記載のブートプログラム切替え装置。

【請求項7】 更に、前記CPUに電力を供給する電力供給手段と、

前記電力供給手段による供給電圧を監視する電圧監視手段とを有し、

前記切替え手段は、前記電圧監視手段により供給電圧の低下が検出されると前記CPUにおけるブートプログラムの論理アドレスに前記既定ブートプログラムの物理アドレスを割り当て、前記リセット手段より前記CPUがリセットされると前記CPUにおけるブートプログラムの論理アドレスに前記保持手段における物理アドレスを順次割り当てることを特徴とする請求項2記載のブートプログラム切替え装置。

【請求項8】 更に、前記CPUがリセットされてからの経過時間を計測する時間計測手段と、

前記CPUが正常にブートされたか否かを判断する判断手段とを有し、

前記判断手段により前記CPUが所定時間以内に正常にブートされなかったと判断された場合、前記リセット手段により前記CPUをリセットすることを特徴とする請求項1記載のブートプログラム切替え装置。

【請求項9】 更に、前記CPUが正常にブートされる手順を監視する監視手段を有し、

前記監視手段により前記手順が所定の手順に等しくない

と判断された場合、前記リセット手段により前記CPUをリセットすることを特徴とする請求項1記載のブートプログラム切替え装置。

【請求項10】 更に、前記ブートプログラムを選択する選択手段を有し、

前記監視手段により前記CPUが正常にブートされた手順が前記選択手段で選択されたブートプログラムによるブートの手順に等しくないと判断された場合、前記リセット手段により前記CPUをリセットすることを特徴とする請求項9記載のブートプログラム切替え装置。

【請求項11】 CPUをブートするためのブートプログラムを複数保持し、

前記CPUが強制リセットされたタイミングで、前記複数のブートプログラムを切り替えることを特徴とするブートプログラム切替え方法。

【請求項12】 前記複数のブートプログラムのうち、1つは通常使用される既定ブートプログラムであることを特徴とする請求項11記載のブートプログラム切替え方法。

【請求項13】 前記CPUにおけるブートプログラムの論理アドレス領域に前記保持手段における物理アドレスを割り当てることにより、前記複数のブートプログラムを切り替えることを特徴とする請求項11記載のブートプログラム切替え方法。

【請求項14】 更に、前記CPUへの電力供給を監視し、

前記CPUへの供給電圧の低下が検出されると前記CPUにおけるブートプログラムの論理アドレスに前記既定ブートプログラムの物理アドレスを割り当て、前記リセット手段より前記CPUがリセットされると前記CPUにおけるブートプログラムの論理アドレスに前記保持手段における物理アドレスを順次割り当てることを特徴とする請求項12記載のブートプログラム切替え方法。

【請求項15】 更に、前記CPUがリセットされてから所定時間以内に正常にブートされなかった場合、前記リセット手段により前記CPUをリセットすることを特徴とする請求項11記載のブートプログラム切替え方法。

【請求項16】 更に、前記CPUが正常にブートされる手順を監視し、

前記手順が所定の手順に等しくないと判断された場合、前記リセット手段により前記CPUをリセットすることを特徴とする請求項11記載のブートプログラム切替え方法。

【請求項17】 更に、前記ブートプログラムを選択し、

前記CPUが正常にブートされた手順が前記選択されたブートプログラムによるブートの手順に等しくないと判断された場合、前記リセット手段により前記CPUをリセットすることを特徴とする請求項16記載のブートプログラム切替え方法。

プログラム切替え方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明はブートプログラム切替え装置及びその方法に関し、例えば複数のブートプログラム領域を備えるCPUのブートプログラム切替え装置及びその方法に関する。

【0002】

【従来の技術】従来のCPUを備えるシステムにおけるブートプログラムは、プログラムROM領域内で、CPUのブートアドレスとして固定された所定の箇所におかれていた。そして、電源ON時あるいはCPUリセット時には、該所定領域からブートプログラムが起動されることにより、CPUは動作を開始していた。

【0003】

【発明が解決しようとする課題】しかしながら、CPUシステムがブートプログラムを複数備えている場合に、例えば通常のブートプログラムが破壊され、別のプログラムROM領域にあるブートプログラムでブートしなければならない時等、ブートプログラムを切り替えようとする場合には、CPUのアドレスバスのマップをハードウェア上で切り替える必要がある。このためには、基板上のスイッチを切り替える等、煩雑な操作が必要であり、特に不慣れなオペレータでは操作ミスが発生する可能性も高い。また、3種以上のブートプログラムを切り替えて使用するような場合には、なおさら操作が複雑となる。

【0004】本発明は上述した課題を解決するためになされたものであり、複数のブートプログラムを操作者の手を煩わせることなく、自動的に切替可能なブートプログラム切替え装置及びその方法を提供することを目的とする。

【0005】

【課題を解決するための手段】上述した目的を達成するための一手段として、本発明は以下の構成を備える。即ち、CPUをブートするためのブートプログラムを複数保持する保持手段と、前記CPUの強制リセットを行うリセット手段と、前記リセット手段によるリセットのタイミングで、前記保持手段に保持された複数のブートプログラムを切り替える切替え手段とを有することを特徴とする。

【0006】例えば、前記保持手段に保持された複数のブートプログラムのうち、1つは通常使用される既定ブートプログラムであることを特徴とする。

【0007】例えば、前記切替え手段は、前記CPUにおけるブートプログラムの論理アドレス領域に前記保持手段における物理アドレスを割り当てることを特徴とする。

【0008】例えば、前記保持手段はROMであることを特徴とする。

【0009】例えば、前記保持手段は複数のROMであり、各ROM毎に1つのブートプログラムを保持することを特徴とする。

【0010】例えば、前記切替え手段は、前記複数のROMのいずれかを特定する信号を出力し、該信号を切り替えることにより複数のブートプログラムを切り替えることを特徴とする。

【0011】更に、前記CPUに電力を供給する電力供給手段と、前記電力供給手段による供給電圧を監視する電圧監視手段とを有し、前記切替え手段は、前記電圧監視手段により供給電圧の低下が検出されると前記CPUにおけるブートプログラムの論理アドレスに前記既定ブートプログラムの物理アドレスを割り当て、前記リセット手段より前記CPUがリセットされると前記CPUにおけるブートプログラムの論理アドレスに前記保持手段における物理アドレスを順次割り当てることを特徴とする。

【0012】更に、前記CPUがリセットされてからの経過時間を計測する時間計測手段と、前記CPUが正常にブートされたか否かを判断する判断手段とを有し、前記判断手段により前記CPUが所定時間以内に正常にブートされなかったと判断された場合、前記リセット手段により前記CPUをリセットすることを特徴とする。

【0013】更に、前記CPUが正常にブートされる手順を監視する監視手段を有し、前記監視手段により前記手順が所定の手順に等しくないとは判断された場合、前記リセット手段により前記CPUをリセットすることを特徴とする。

【0014】更に、前記ブートプログラムを選択する選択手段を有し、前記監視手段により前記CPUが正常にブートされた手順が前記選択手段で選択されたブートプログラムによるブートの手順に等しくないとは判断された場合、前記リセット手段により前記CPUをリセットすることを特徴とする。

【0015】

【作用】以上の構成により、CPUのブートプログラムを切り替える必要が生じた場合に、装置内で自動的に通常の領域外にあるブートプログラムを検出し、動作させる事ができる。

【0016】

【実施例】以下、本発明に係る一実施例について、図面を参照して詳細に説明する。

【0017】＜第1実施例＞図1に、本実施例におけるCPUのブートプログラム切替え装置の構成を示す図である。尚、図1においては本実施例に係る構成のみを抜粋しており、例えばCPUの作業領域となるRAM等の他の構成が追加されていても良い。

【0018】図1において、CPU101はプログラムROM（以下、ROMと称する）104a～104dに格納されたブートプログラムより起動される。ROM1

04a~104dにおいて、ブートプログラムのアドレスはアドレス信号108の下位により接続される。尚、アドレス信号108は、実際にブートされるブートプログラムのアドレス（ブートアドレス）を示しているものとする。

【0019】アドレスデコーダ103は、アドレス信号108の上位をデコードし、チップセレクト信号112a~112dを生成する。CPU101がアクセスする際にROM104a~104dのいずれか1つ、あるいはデータバスのアクセス単位を形成する一組が、チップ

セレクト信号112a~112dでセレクトされる。例えば、チップセレクト信号112aによりROM104aがセレクトされる。尚、本実施例においては、通常、例えば電源ON時にはROM104aにあるブートプログラムによりCPU101は動作を開始するものとし、以下、ROM104aに格納されているブートプログラムを「既定プログラム」と称する。

【0020】リセット部102は、電源電圧及びリセットスイッチ106を監視して、電源電圧低下時あるいはリセットスイッチ106が操作された場合に、所定の幅のリセットパルスを発生する。このリセット信号はCPUリセット信号109、及びアドレスデコーダ103のチップセレクトを制御するアドレスデコーダリセット信号110、111として作用する。アドレスデコーダリセット信号110は電源監視により発生するリセット信号であり、アドレスデコーダリセット信号111はリセットスイッチ106を動作させた場合のリセット信号である。

【0021】通常の電源ON時には、リセット部102はCPUリセット信号109及びアドレスデコーダリセット信号110を発生する。アドレスデコーダリセット信号110が発生し、CPU101がリブートした場合に、アドレスデコーダ103はCPU101の出力したアドレス信号108をそのままデコードしてチップセレクト信号112a~112dを発生する。ここで、ROM104a~104dのアドレスが、ROM104a、ROM104b、ROM104c、ROM104dの順で「0」から順にマップされているとする。すると、通常CPU101はアドレス=「0」からアクセスしてブートするため、ブート直後はチップセレクト信号112aがアサートされ、即ちROM104aが選択されることにより既定プログラムが起動され、通常のブート動作が行われる。

【0022】一方、リセットスイッチ106が操作されたときには、リセット部102はCPUリセット信号109及びアドレスデコーダリセット信号111を発生する。これによりCPU101は上述した電源ON時と同様にリブートする。しかしながら、この場合アドレスデコーダ103はCPU101が発生するアドレスを変換して、通常のブートROM104aの次にマップされて

いるROM104bのチップセレクト信号112bを発生する。これにより、CPU101は通常のブートプログラム以外の他のブートプログラムによる動作を試みる。

【0023】以上説明したように、リセットスイッチ106を操作することによりアドレスデコーダリセット信号111が発生する毎に、アドレスデコーダ103はブートアドレスのチップセレクト信号112a~112dを順次切り替えて行く。

【0024】図1において105は正常ブート監視部であり、CPU101のI/Oポート等と接続されている。正常ブート監視部105は、CPU101のブート直後に、CPU101が所定の手順のソフトウェア操作で正常ブート監視部105にアクセスしたか否かを監視する。そして、それに対応して表示LED107の明滅を操作することにより、CPU101が正常にブートしたか否かを操作者に報知する。操作者はCPU101が正常にブートしなかったことを認識すると、リセットスイッチ106を押下する。すると、アドレスデコーダ103はリセットスイッチの操作に対応して上述したようにチップセレクト信号112a~112dを切り替えることにより、ブートプログラム領域（ROM104a~104d）を切り替える。即ち、CPU101におけるブートプログラムの論理アドレスに、ROM104a~104dの物理アドレスを順次割り当てる。

【0025】以下、図2にリセット部102及びアドレスデコーダ103の詳細構成を示す。図2において、上述した図1と同一の構成要素には同一番号を付し、説明を省略する。

【0026】図2において、リセット部102は電源監視部201とスイッチリセット部202、及びORゲート203とから成る。電源監視部201は、電源ON時及び電源電圧が所定値より下がった場合に、所定時間の負のリセットパルスをアドレスデコーダリセット信号110として発生する。同様に、リセットスイッチ部202はリセットスイッチ106が操作された場合に、所定時間の負のリセットパルスをアドレスデコーダリセット信号111として発生するものとする。ORゲート203は、アドレスデコーダリセット信号110、111の論理和を取り、CPU101をリセットするリセット信号109を発生する。

【0027】一方、アドレスデコーダ103は、カウンタ204、デコーダ205、ゲート部206とから成る。ここで、カウンタ204は2ビットのカウンタ値を有し、デコーダ205は2ビットのデコーダであると仮定する。デコーダ205は、カウンタ204の出力に応じて4ビット出力のいずれかをアサートする。また、ゲート部206はCPU101から入力されるアドレス信号108に応じて、チップセレクト信号112a~112dのいずれかをアサートする。

【0028】図3に、図2における各信号のタイミングチャートを示す。図3に示すタイミングチャートでは、原点において電源をOFFからONに切り替えたとする。すると、電源監視リセット信号110は、電源電圧が上昇して所定電圧に達した時点から、所定時間だけ負のパルスを発生する。この後、CPU101はブートし、所定のプログラムアドレスから動作を開始する。尚、通常、所定のプログラムアドレスはアドレス「0」であり、アドレス「0」に相当するROM104aのチップセレクト信号は112aである。

【0029】リセット信号110は、カウンタ204をクリアして、カウンタ204からの出力信号207を「0」にして、デコーダ205の出力信号のうち、208aをアサートする。

【0030】電源ONの直後、所定時間以内にCPU101がブートしない場合、操作者はLED107でそれを検出し、リセットスイッチ106を操作する。すると、スイッチリセット部202はアドレスデコーダリセット信号111に所定の時間幅の負のパルスを発生し、これはリセット信号109を介してCPU101をブートするとともに、カウンタ204のクロックに入力されてカウンタ204をカウントアップする。

【0031】カウンタ204においてカウントアップされた出力207に応じて、デコーダ205はその出力をアサートする。即ち、図3に示すように、カウンタ出力207が「0」である場合にはデコーダ205は出力208aをアサートし、カウンタ出力207が「1」である場合には出力208bをアサートする。同様に、カウンタ出力207が「3」であれば出力208c、「4」であれば208dがアサートされる。

【0032】そして、ゲート部206において、CPU101から入力されるアドレス信号108及びカウンタ204からの出力207に応じて、チップセレクト信号112a～112dのいずれかがアサートされる。例えば、デコーダ205の出力208aが「1」であれば、アドレス信号108が「0」である場合にはゲート部206からの出力としてチップセレクト信号112aがアサートされる。またこのとき、アドレス信号108が「1」である場合にはチップセレクト信号112bが、アドレス信号108が「2」及び「3」である場合には、それぞれチップセレクト信号112c、112dがアサートされる。これは正常ブート状態で正常なROMアクセスに相当する。

【0033】また、デコーダ205の出力208bが「1」である場合はブート時、即ちアドレス信号108が「0」の状態ではチップセレクト信号112bをアサートする。同様にデコーダ205の出力208cもしくは208dが「1」である場合には、アドレス信号108が「0」のブート時にチップセレクト信号112c、112dをアサートする。

【0034】このように、カウンタ204においてカウントされた出力207に応じて、アドレス「0」に対応したチップセレクト信号112aから順に、チップセレクト信号112b、112c、112dと、順次切り替えられる。

【0035】以上説明した様に本実施例によれば、CPUが所定時間内にブートされるまでブートプログラムの格納アドレスを順次変更していくことができる。従って、煩雑な操作を行うことなく、他のブートプログラムからの起動が行える。

【0036】尚、本実施例では既定ブートプログラムがROM104aに格納されている場合について説明を行ったが、本発明はこの例に限られるものではなく、どのROMに格納されているブートプログラムを既定プログラムとしてもよい。この場合、図2に示すゲート部206の構成を、アドレス信号108が「0」である場合にアサートされるチップセレクト信号が112a以外になるように変更すれば良い。

【0037】また、ブートプログラムを格納するROMが複数である場合について説明を行ったが、例えば単一のROM内で異なる複数のアドレスにブートプログラムを格納し、カウンタ出力207に応じてCPUからのアドレス信号108が「0」である場合の物理アドレスマップを変更するように構成してもよい。

【0038】＜第2実施例＞以下、本発明に係る第2実施例について説明する。

【0039】図4に、第2実施例におけるCPUのブートプログラム切替え装置の構成を示す。図4において、上述した第1実施例における図1と同様の構成については同一番号を付し、説明を省略する。

【0040】第2実施例において、CPU401はウォッチドッグタイマ403を内蔵している。ここで、ウォッチドッグタイマ403は、CPU401のクロックで動作するタイマである。ウォッチドッグタイマ403は、所定時間以内にソフトウェア操作によるリセットが行なわれない場合には、CPU暴走と判断してCPU401の動作をリセットするとともに、リセット信号111を出力する。

【0041】リセット部402は、上述した図2に示したリセット部102と同様に電源電圧及びリセットスイッチ106を監視するが、両者の出力は分離していないことを特徴とする。

【0042】第2実施例においては、電源ON直後は上述した第1実施例と同様にCPU401をリセットしてブートさせるとともに、アドレスデコーダ103内部のカウンタ204をクリアする。

【0043】また、ウォッチドッグタイマ403は、所定時間内にリセットされない場合にCPU401をリセットするとともに、リセット信号111を発生させ、アドレスデコーダ103のカウンタ204をカウントアップ

ブする。そして、その結果セレクトされたROMでも正常にブートしなかった場合、ウォッチドッグタイマ403は所定時間間隔でリセット信号111を出力し続け、アドレスデコーダ103はROM104a~104dの選択を切り替えてゆき、正常にブートするプログラムROMを自動的に検出する。

【0044】以上説明したように第2実施例によれば、操作者によるリセットスイッチの押下が無くても、所定時間経過後にブートできなければ、自動的に次のブートプログラムが格納されているアドレスに切り替えることができる。

【0045】<第3実施例>以下、本発明に係る第3実施例について説明する。

【0046】図5に、第3実施例におけるCPUのブートアドレス変換部の構成を示す。図5において、上述した第1実施例における図1と同様の構成については同一番号を付し、説明を省略する。

【0047】第3実施例においては、上述した第1実施例及び第2実施例とを組み合わせた効果を有することを特徴とする。

【0048】図5において、正常ブート監視部501はタイマ502でCPU101のブート直後の所定時間以内に、CPU101からの所定の手順でアクセスを監視する。そして、CPU101から所定の手順でのアクセスがない場合にはCPU暴走であると判断し、リセット信号を発生する。該リセット信号はゲート503でリセットスイッチ106からの信号と合成され、リセット部102に入力される。

【0049】リセット部102では、正常ブート監視部501がリセット信号を発生させる毎にリセット信号111を発生させ、アドレスデコーダ103内部のカウンタ204をカウントアップする。これにより、上述した第2実施例と同様に、タイマ502で設定された時間間隔で正常にブートするブートプログラムを検索する。

【0050】以上説明したように第3実施例によれば、上述した第1実施例及び第2実施例を組み合わせることにより、操作者によるリセットスイッチの押下があるか、またはリセットスイッチの押下が無くても所定時間経過後にブートできなければ、次のブートプログラムが格納されているアドレスに切り替えることができる。

【0051】<第4実施例>以下、本発明に係る第4実施例について説明する。

【0052】図6に、第4実施例におけるCPUのブートプログラム切替え装置の構成を示す。図6において、上述した第1実施例における図1と同様の構成については同一番号を付し、説明を省略する。

【0053】図6においては、ブートプログラム選択スイッチ601で、操作者がブートプログラムを選択することができる。ブートプログラム選択スイッチ601で選択されるブートプログラムへのアクセス手順は、正常

ブート監視部501内では、予めアクセスパターンテーブルとして備えられている。正常ブート監視部501は、上述した第3実施例と同様にブート後所定時間以内にCPU101から所定手順のアクセスの有無を検出するが、更に、CPU101からのアクセス手順がブートプログラム選択スイッチ601で選択されたアクセスパターンであるか否かも検出する。

【0054】正常ブート監視部501において、CPU101から正常なアクセスが行われなかったと判断される場合、例えば、ブートプログラム選択スイッチ601で選択されたアクセス手順と異なるアクセスが行われた場合、正常ブート監視部501はORゲート503へリセット信号を発生し、上述した第3実施例同様に、次のブートプログラムが格納されているROMを選択する。

【0055】以上説明したように第4実施例によれば、予めブートプログラム毎にそれを識別するアクセスパターンを記憶しておくことにより、容易に任意のブートプログラムを動作させることができる。

【0056】尚、上述した第1~第4実施例においては、ブートプログラムが4つである例について説明を行ったが、本発明はもちろんこの限りではなく、2つ以上のブートプログラムが存在する場合に適用可能である。

【0057】尚、本発明は、複数の機器から構成されるシステムに適用しても、1つの機器から成る装置に適用しても良い。また、本発明はシステム或は装置にプログラムを供給することによって達成される場合にも適用できることはいうまでもない。

【0058】

【発明の効果】以上説明したように本発明によれば、複数のブートプログラム領域を持つCPUシステムにおいて、複雑な操作を行うことなくブートプログラム領域を変更することが可能となる。

【0059】

【図面の簡単な説明】

【図1】本発明に係る一実施例におけるCPUのブートプログラム切替え装置の構成を示すブロック図である。

【図2】本実施例におけるリセット部及びアドレスデコーダの詳細構成を示す図である。

【図3】本実施例のリセット部及びアドレスデコーダにおけるタイミングチャートである。

【図4】本発明に係る第2実施例におけるCPUのブートプログラム切替え装置の構成を示すブロック図である。

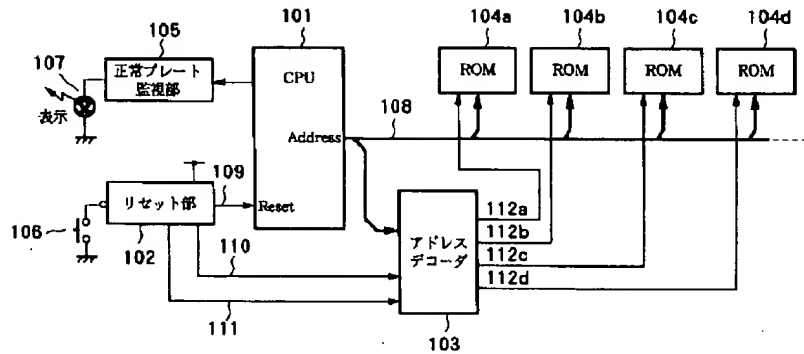
【図5】本発明に係る第3実施例におけるCPUのブートプログラム切替え装置の構成を示すブロック図である。

【図6】本発明に係る第4実施例におけるCPUのブートプログラム切替え装置の構成を示すブロック図である。

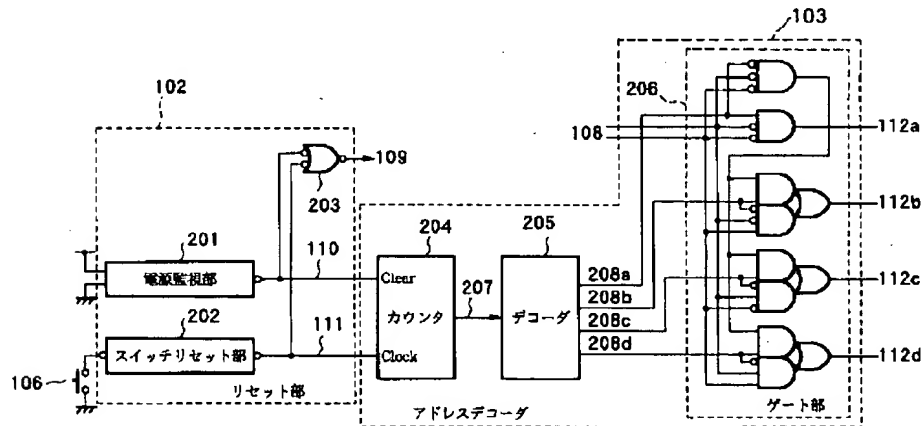
【符号の説明】

- | | |
|--|---|
| <p>11</p> <p>104, 401 CPU</p> <p>102 リセット部</p> <p>103 アドレスデコーダ</p> <p>104a~104d ROM</p> <p>105, 501 正常ブート監視部</p> <p>106, 402 リセットスイッチ、</p> | <p>12</p> <p>201 電源監視リセット部</p> <p>202 スイッチリセット部</p> <p>204 カウンタ</p> <p>205 デコーダ</p> <p>206 ゲート部</p> <p>601 ブートプログラム選択スイッチ</p> |
|--|---|

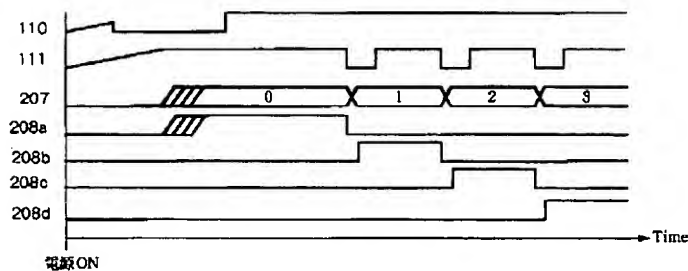
【図1】



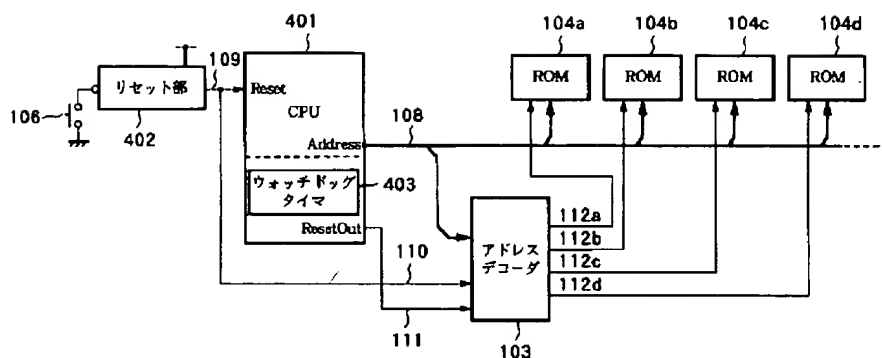
【図2】



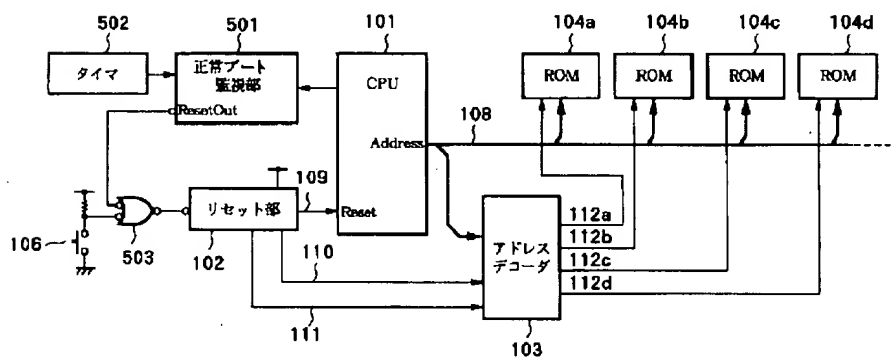
【図3】



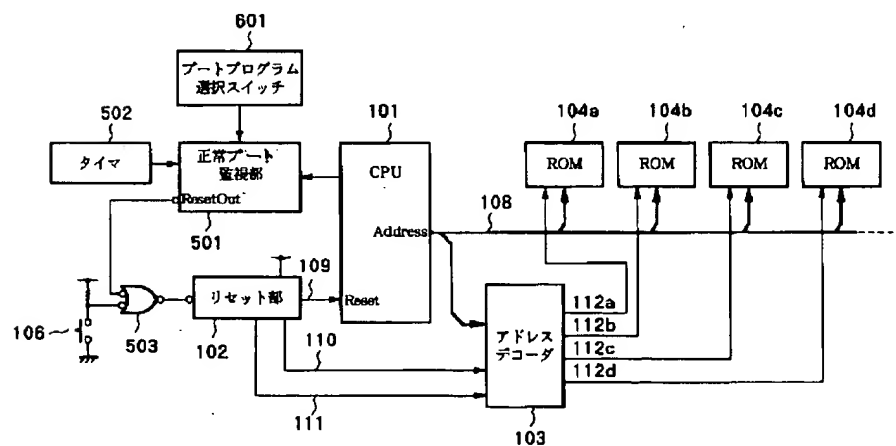
【図4】



【図5】



【図6】



Linux Remote-Boot mini-HOWTO: